Soft Computing (2021) 25:11245–11267
https://doi.org/10.1007/s00500-021-05894-z

APPLICATION OF SOFT COMPUTING

Application of hybrid binary tournament-based quantum-behaved particle swarm optimization on an imperfect production inventory problem

Nirmal Kumar[1] · **Amalesh Kumar Manna**[1] · **Ali Akbar Shaikh**[1] · **Asoke Kumar Bhunia**[1]

Accepted: 18 May 2021 / Published online: 14 June 2021

Abstract
Nowadays, use of various types of hybrid metaheuristic algorithms attracts the researchers to optimize the average profit or cost of an inventory system to avoid the local optimality due to high nonlinearity of the corresponding optimization problem. This paper deals with an application of binary tournament-based quantum-behaved particle swarm optimization algorithms on an imperfect production inventory problem with shortages. In order to reduce the production of defective items, modern/improvement technology has been incorporated in the production system. Also, the demand of the product is assumed to be dependent on its warranty period and selling price. The main objective of this study is to optimize the production rate, production period, selling price of the product, manufacturer's improvement technology level and maximum shortage level as well as maximize the average profit of the production system. For this purpose, three hybrid metaheuristic algorithms based on binary tournamenting and different variants of quantum-behaved PSO techniques have been developed. Then to examine the validity of the proposed model, three numerical examples have been solved. Considering each example, nonparametric statistical tests have been performed by using four different methods to analyze the performance of the used algorithms. Finally, sensitivity analyses have been performed to investigate the effects of different parameters on optimal policy.

Keywords Imperfect production · Partial backlogging · Dynamic demand · Binary tournamenting QPSOs

1 Introduction

In any manufacturing firm, all the produced items are not perfect due to imperfect production process or other factors. Considering this realistic situation of production, several researchers have developed various production models and reported in the existing literature. In 1986, Rosenblatt and Lee (1986) first introduced the concept of imperfect production in manufacturing system. Salameh and Jaber (2000) formulated an imperfect production problem. Sana et al. (2007) uplifted a production problem with volume flexible cost under imperfect production system. Sarker and Moon (2011) presented a model corresponding to the imperfect production system with development cost investment. Some other interesting research works related to imperfect production process were found in in the works of Chiu (2003), Goyal and Cárdenas-Barrón (2005), Modak et al. (2015), Das et al. (2017), Manna et al. (2017a), Mallick et al. (2018), among others. Jain et al. (2018) introduced repairing concept in the imperfect production system under fuzzy environment. Taleizadeh et al. (2019) examined product quality and returns in an imperfect production system under two warranty policies. Manna et al. (2019) established two-plant production model for two quality items under fuzzy environment. Rahman et al. (2020) developed a production inventory model with credit-linked demand in interval environment. They also assumed the produced products which deteriorate with time. Shaikh et al. (2020) developed an EPQ model with partial trade credit policy- and price-dependent demand for deteriorating items. Mishra et al. (2021) proposed the concept of

✉ Ali Akbar Shaikh
aakbarshaikh@gmail.com

Nirmal Kumar
kumarnirmal843@gmail.com

Amalesh Kumar Manna
akmanna1987@gmail.com

Asoke Kumar Bhunia
akbhunia@math.buruniv.ac.in

1 Department of Mathematics, The University of Burdwan, Burdwan 713104, India

preservation technology in a production inventory model to protect the items from deterioration.

Sometimes, it is observed that stock-out situation arises in integrated production system due to uncertain demand, offering of discount facility, deterioration effect of produced items, etc. As a result, in stock-out situation, manufacturer cannot able to fulfill the demand of customers/retailers. During stock-out period, two situations may occur: (i) all the customers are willing to wait for receiving the product and (ii) a part of the customers only are willing to wait for receiving the product. The second case of situation is known as partial backlogging. Abad (2000) developed a lot-size problem with partial backordering for perishable items. An inventory model for deteriorating items was developed by Giri et al. (2003). Ouyang and Chang (2013) proposed an optimal production model with complete backlogging and permissible delay in payments. In interval environment, a partially integrated production model with variable demand and partial backordering was introduced by Bhunia et al. (2017). Shaikh et al. (2017) established an inventory model for non-instantaneous deteriorating items with price- and stock-dependent demand under partially backlogged situation. Tiwari et al. (2018) investigated a green production problem with partial backordering for multi-items. Das et al. (2020) proposed an inventory model with partially backlogging and price-dependent demand for deteriorating items considering preservation facilities. Later, Das et al. (2021) developed an inventory model with partial backlogged shortages and trade credit financing under preservation technology for deteriorating items via particle swarm optimization. Apart from the earlier mentioned works, several researchers, viz. Jamal et al. (1997), Chiu (2003), Chen and Lo (2006), Chakraborty et al. (2013), studied different inventory models with complete backlogging/ partial backlogging.

The classical inventory model was developed under the assumption constant demand. After that, a number of researchers reported various types of customer's/retailer's demand dependent on several factors such as Shaikh et al. (2019)(stock-dependent demand), Giri et al. (2003) (ramp-type demand), Jain et al. (2018) (time-dependent demand), Jaggi et al. (2017) (price discount demand), Lee and Yao (1998) (fuzzy demand), Manna et al. (2017a) (advertisement-dependent demand), Dye and Yang (2015) (credit-linked demand), etc. However, it is very difficult to estimate the market demand due to the lack of historical data. The warranty period is an vital issue to take the customers' decision for purchasing the product. Yeh et al. (2005) proposed warranty policy for repairable items. Wu et al. (2009) optimized price, warranty length, production rate in a production inventory model. Wang and Sheu (2003) considered free warranty policy in their production model. Chung (2013) considered production model where the demand is dependent on warranty period of the product. Taleizadeh et al. (2017) introduced warranty policy in a supply chain model. Recently, Manna et al. (2020) investigated the effects of warranty period and selling price of the product on customers' demand in a manufacturing system. In Table 1, a comprehensive review of related articles reported in the literature is presented.

To optimize the average profit/cost of an inventory model, different methods can be applied such as,

(i) Direct search method
(ii) Gradient-based method
(iii) Metaheuristic method

However, in the proposed work, the optimization (maximization) problem corresponding to the proposed production inventory model is highly nonlinear in nature and non-concave. So, this optimization problem cannot be solved by traditional direct and gradient-based optimization methods. These methods have some limitations. Among these limitations, one is that the traditional nonlinear optimization methods very often stuck to the local optimum. So, authors are bound to choose metaheuristic methods. All metaheuristic algorithms have been developed from the activities of the social organisms, properties of environments, properties of some instruments of physics, etc. Over the previous few decades, various nature-inspired algorithms have been proposed such as genetic algorithm (Goldberg 2006), particle swarm optimization (Eberhart and Kennedy 1995; Clerc and Kennedy 2002; Sun et al. 2005, 2011; Xi et al. 2008; Coelho 2010), krill herd algorithm (Abualigah 2019), grasshopper optimization algorithm (Abualigah and Diabat 2020), arithmetic optimization algorithm (Abualigah et al. 2021a), sine cosine algorithm (Abualigah and Diabat 2021b), differential evolution algorithm (Storn and Price 1997), tournament differential evolution algorithm (Akhtar et al. 2020). Besides, some modified versions of these algorithms are proposed (Duary et al. 2020; Kumar et al. 2019, 2020, 2021a, b). Surprisingly, PSO and its modified versions have been confoundedly used to optimize the average profit/cost of an inventory model in the recent years.

Particle swarm optimization (PSO) was proposed by Eberhart and Kennedy (1995). Later, Clerc and Kennedy (2002) modified the original PSO algorithm by inserting a constriction factor. Since then, the corresponding PSO is known as PSO-Co. Thereafter, Sun et al. (2004) proposed a modified PSO algorithm known as quantum-behaved PSO (QPSO) which is based on the quantum behavior of the particles. Then, to accelerate the performance of QPSO, Xu and Sun (2005) developed adaptive QPSO (AQPSO), Xi et al. (2008) developed Weighted QPSO (WQPSO), Coelho (2010) developed Gaussian QPSO (GQPSO), Kumar et al. (2019) developed AGQPSO, etc. In the current work, only three embedded QPSO algorithms, AQPSO, GQPSO and

Table 1 Summary of the literature, based on various assumptions related to the proposed model

Author(s) with year	Development technology-dependent defective rate	Demand rate dependent on		Production cost dependent on			Inventory and waiting time-dependent backlogging rate	Optimization technique
		warranty period	selling price	production rate	warranty period	development technology		
Abad (2000)	×	×	×	×	×	×	×	Analytical method
Bhunia et al. (2017)	×	×	✓	✓	×	×	✓	PSO-CO algorithm
Chakraborty et al. (2013)	×	×	×	×	×	×	×	IFP technique
Coelho (2010)	×	×	×	×	×	×	×	GQPSO
Kumar et al. (2019)	×	×	×	×	×	×	×	Tournament AGQPSO
Sana et al. (2007)	×	×	×	✓	×	×	×	Analytical method
Sarker and Moon (2011)	✓	×	×	✓	×	×	×	Analytical method
Shaikh et al. (2019)	×	×	✓	×	×	×	×	Analytical method
Jaggi et al. (2017)	×	×	✓	×	×	×	×	Analytical method
Taleizadeh et al. (2017)	×	✓	✓	×	×	×	×	NSGA-II
Xu and Sun (2005)	×	×	×	×	×	×	×	QPSO
Yeh et al. (2005)	×	×	×	×	✓	×	×	Analytical method
Wu et al. (2009)	×	✓	✓	×	×	×	×	Analytical method
Present paper	✓	✓	✓	✓	✓	✓	✓	Tournament QPSOs

AGQPSO algorithms, are used to solve the optimization problem. Apart from these algorithms, by using these algorithms, we have developed three different algorithms based on the concept of binary tournamenting process which is followed in a game. These are called as T2-AQPSO, T2-GQPSO and T2-AGQPSO. Finally, the results are compared with each other and said PSOs.

In this paper, an imperfect production inventory model with partial backlogging and dynamic demand has been developed. Here production of defective items has been reduced by considering modern/improvement technology. Also, the partial backorder rate is dependent on the length of the waiting time of the customers. Furthermore, the demand rate of the customers is assumed to be dependent on warranty period and selling price of the product. Then, six metaheuristic optimization techniques AQPSO, GQPSO, AGQPSO, T2-AQPSO, T2-GQPSO and T2-AGQPSO have been used for solving the corresponding maximization problem (average profit) of the proposed model and compared the results obtained.

The leftover of this work is constructed as follows. The next section presents notation and assumptions for formulating the production inventory model. Mathematical formulation of the proposed model is provided in Sect. 3. Section 4 demonstrates the solution methodology to determine the optimal/best found solutions. Numerical experiments and sensitivity analyses are shown in Sects. 5 and 6, respectively. Managerial insights and conclusions are drawn in Sects. 7 and 8, respectively. Finally, limitations with future research scope of this work are presented in Sect. 9.

2 Notation and assumptions

The following notation and assumptions have been considered throughout the manuscript.

2.1 Notation

2.2 Assumptions

(i) The production system produces single item and time horizon is infinite.

(ii) The production system produces perfect item at the rate $(1-\theta)P$ where $0 < \theta << 1$. During the production period $(0, t_p)$, the manufacturing company will invest modern/improvement technology cost to reduce the defective production. The rate of defectiveness of produced items is defined as follows:

$$\theta(\eta) = \theta_0 e^{-\xi\eta},\ \eta \in [0, +\infty) \tag{1}$$

Decision Variables		
P	:	Production rate
t_p	:	Duration of production run time
η	:	Manufacturer's improvement technology level to reduce the production of defective units
s	:	Selling price per unit for perfect quality item
Q_s	:	Maximum shortage level
Dependent Variable		
T_1	:	Time point when the inventory level reaches zero
T_2	:	Time point when production starts to back-logged
T	:	Time point of business period
Others		
$I(t)$	:	Inventory level of perfect quality items
$\theta(\eta)$	:	Reduced defective production rate, a decreasing function with respect to η
θ_0	:	Defective production rate without development technology
ω_p	:	Warranty period of the product which is sold to the customer
$C(P,\eta)$	:	Production cost per unit
$D(\omega_p, s)$	:	Demand rate of the retailer
α	:	Partial backorder rate
w_p	:	Warranty period of the product
h	:	Holding cost/ unit / unit time
A	:	Setup cost/ cycle
c_b	:	Backordering cost/ unit / unit time
N_p	:	Swarm size
Max_gen	:	Maximum number of generations

which is a decreasing function with respect to η for suitable value of $\xi > 0$. The graphical representation of defective rate vs modern/improvement technology is given in Fig. 1.

(iii) The demand of an item is dependent on warranty period along with selling price, and mathematically it can be represented as follows:

$$D(\omega_p, s) = \beta_1 + \lambda_1\omega_p - \lambda_2 s, \tag{2}$$

where β_1 is a fixed demand of customers and λ_1, λ_2 are coefficients of sensitivity of the customers about warranty period and selling price of the product.

(iv) The unit production cost is a function of production rate and modern/improvement technology, which is given by

$$C(P,\eta) = C_0 + C_1P^{\lambda} + \frac{C_2}{P^{\mu}} + C_3\eta^{\delta} + C_4\omega_p^{\gamma} \tag{3}$$

where C_0 is the fixed production cost, C_1 and C_2 are coefficients of sensitivity of production cost. Also, C_3 and C_4 are coefficients of sensitivity of warranty cost and development cost, respectively.

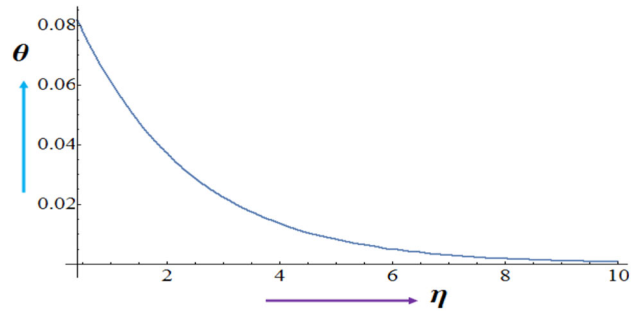

Fig. 1 Pictorial representation of defective rate vs η

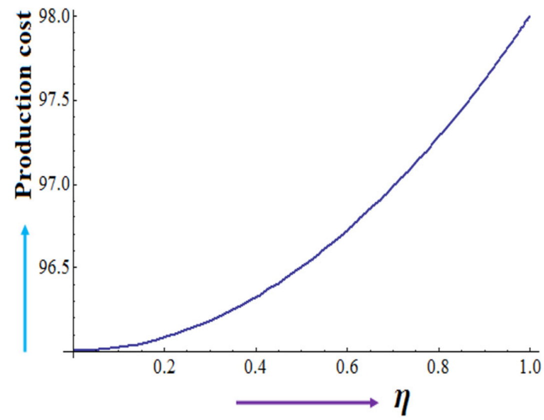

Fig. 2 Production cost vs η

The geometrical representation of unit production cost with respect to η and P is given in Figs. 2 and 3, respectively.

(v) The warranty cost (c_w) is dependent on warranty period (ω_p) and is given by

$$c_w(\omega_p) = a + b\omega_p \tag{4}$$

where a is a fixed warranty cost and b is the coefficient of sensitivity of the warranty period.

(vi) During the stock-out period, some of the customers are willing to wait for receive the product. Here, the backlogging rate is in the form $\frac{1}{1+\alpha\left\{T_2-t-\frac{I(t)}{(1-\theta)P}\right\}}$, where α is the parameter of backlogging rate.

3 Mathematical formulation of the proposed model

Let us assume that a manufacturing firm starts the production at time $t = 0$ and continues up to the time $t = t_p$ with the production rate P. During the production time, manufacturing firm produces some defective items along with perfect

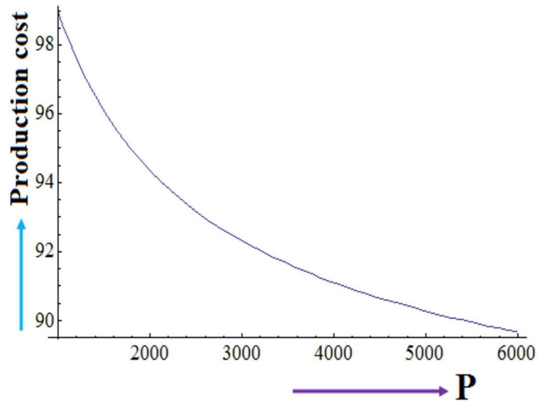

Fig. 3 Production cost vs P

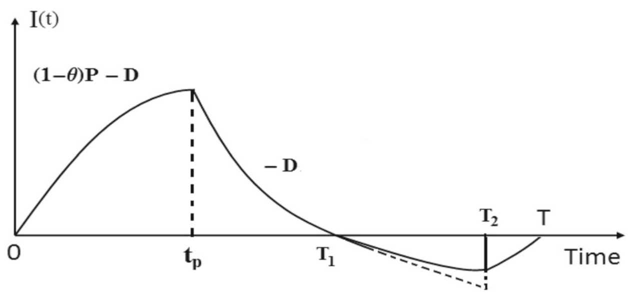

Fig. 4 Pictorial representation of the inventory level

ones. Here, it has been considered that the production firm produces perfect items with the rate $(1-\theta)P$ up to the time $t = t_p$ which satisfy the customers' demand with the rate $D(\omega_p, s)$. The rest of the produced items are stored in the store room with the rate $(1-\theta)P - D(\omega_p, s)$ up to the time $t = t_p$. After that, the production process stops and the inventory level depletes gradually during the time interval $[t_p, T_1]$ due to customers' demand only. Thereafter, shortages occur and continue up to the time $t = T_2$. At time $t = T_2$, again production process starts and fulfills the backlogged quantity after meeting up the customers' demand.

Therefore, the inventory level satisfies the governing differential equations as follows:

$$\frac{dI(t)}{dt} = (1-\theta)P - D(\omega_p, s), \quad 0 \le t \le t_p \tag{5}$$

$$\frac{dI(t)}{dt} = -D(\omega_p, s), \quad t_p < t \le T_1 \tag{6}$$

$$\frac{dI(t)}{dt} = -\frac{D(\omega_p, s)}{1 + \alpha\left\{T_2 - t - \frac{I(t)}{(1-\theta)P}\right\}}, \quad T_1 < t \le T_2 \tag{7}$$

$$\frac{dI(t)}{dt} = (1-\theta)P - \frac{D(\omega_p, s)}{1 - \frac{\alpha I(t)}{(1-\theta)P}}, \quad T_2 < t \le T \tag{8}$$

subject to the conditions that $I(0) = 0$, $I(T_1) = 0$, $I(T_2) = -Q_s$ and $I(T) = 0$.

Using the conditions $I(0) = 0$, $I(T_1) = 0$ and $I(T) = 0$, the solutions of the equations $(5) - (8)$ are, respectively, as follows:

$$I(t) = \Big\{(1-\theta)P - D(\omega_p, s)\Big\}t, \quad 0 \le t \le t_p \tag{9}$$

$$I(t) = D(\omega_p, s)(T_1 - t), \quad t_p < t \le T_1 \tag{10}$$

$$\begin{aligned} &\alpha I(t) + \Big[(1-\theta)P\Big\{1 + \alpha(T_2 - T_1)\Big\} - D(\omega_p, s)\Big] \\ &exp\Big\{\frac{\alpha I(t)}{D(\omega_p, s)}\Big\} = \Big[(1-\theta)P\Big\{1 + \alpha(T_2 - t)\Big\} \\ &D(\omega_p, s)\Big], \quad T_1 < t \le T_2 \end{aligned} \tag{11}$$

$$\begin{aligned} &I(t) - \frac{D(\omega_p, s)}{\alpha} log\left|\frac{\Big\{(1-\theta)P - D(\omega_p, s) - \alpha I(t)\Big\}}{\Big\{(1-\theta)P - D(\omega_p, s)\Big\}}\right| \\ &= (1-\theta)P(t - T), \quad T_2 < t \le T \end{aligned} \tag{12}$$

The continuity condition of $I(t)$ at $t = t_p$ implies

$$T_1 = \frac{P}{D(\omega_p, s)}(1-\theta)t_p \tag{13}$$

Again, the condition $I(T_2) = -Q_s$ implies

$$\begin{aligned} T_2 = T_1 + &\frac{1}{\alpha(1-\theta)P}\Big[\Big\{(1-\theta)P - D(\omega_p, s) + \alpha Q_s\Big\} \\ &exp\Big\{\frac{\alpha Q_s}{D(\omega_p, s)}\Big\} - (1-\theta)P + D(\omega_p, s)\Big] \end{aligned} \tag{14}$$

The continuity condition of $I(t)$ at $t = T_2$ implies

$$\begin{aligned} T = T_2 + &\frac{1}{(1-\theta)P}\Bigg[Q_s + \frac{D(\omega_p, s)}{\alpha} \\ &log\left|\frac{\Big\{(1-\theta)P - D(\omega_p, s) + \alpha Q_s\Big\}}{\Big\{(1-\theta)P - D(\omega_p, s)\Big\}}\right|\Bigg] \end{aligned} \tag{15}$$

$$\begin{aligned} &\text{Production cost (PC)} = C(\eta, \omega_p)\Big[\int_0^{t_p} P\,dt + \int_{T_2}^{T} P\,dt\Big] \\ &= \Big(C_0 + C_1 P^{\lambda} + \frac{C_2}{P^{\mu}} + C_3\eta^{\delta} + C_4\omega_p^{\gamma}\Big)P(t_p + T - T_2) \\ &\text{Holding cost (HC)} = h\Big[\int_0^{t_p} I(t)\,dt + \int_{t_p}^{T_1} I(t)\,dt\Big] \\ &= \frac{h}{2}\Big[(1-\theta)Pt_p^2 + D(\omega_p, s)T_1^2 - 2D(\omega_p, s)t_pT_1\Big] \\ &\text{Backorder cost (BC)} = c_b\Big[\int_{T_1}^{T_2} -I(t)\,dt + \int_{T_2}^{T} -I(t)\,dt\Big] \\ &\text{Warranty cost (WC)} = (a + b\omega_p)\vartheta(1-\theta)P \\ &\text{Sales revenue (SR)} = s(1-\theta)P(t_p + T - T_2) \end{aligned} \tag{16}$$

The total profit of the manufacturer can be calculated as follows:

$$
\begin{aligned}
TP(P, t_p, \eta, s, Q_s) &= \text{SR-PC-HC-BC-WC-A} \\
&= s(1-\theta)P(t_p + T - T_2) \\
&\quad -\Big(C_0 + C_1 P^{\lambda} + \frac{C_2}{P^{\mu}} + C_3\eta^{\delta} \\
&\quad +C_4\omega_p^{\gamma}\Big) P(t_p + T - T_2) \\
&\quad -\frac{h}{2}\Big[(1-\theta)Pt_p^2 + D(\omega_p, s)T_1^2 \\
&\quad -2D(\omega_p, s)t_p T_1\Big] \\
&\quad -(a + b\omega_p)\vartheta(1-\theta)P \\
&\quad -c_b\Big[\int_{T_1}^{T_2} -I_3(t)\,dt \\
&\quad +\int_{T_2}^{T} -I_4(t)\,dt\Big] - A
\end{aligned} \tag{17}
$$

Therefore, the average profit of the manufacturer is given by

$$
\Pi(P, t_p, \eta, s, Q_s) = \frac{TP(P, t_p, \eta, s, Q_s)}{T} \tag{18}
$$

Hence, the objective is to determine the optimal production period (t_p^*), production rate (P^*), selling price (s^*), maximum shortage level (Q_s^*) and development technology level (η^*) by maximizing the manufacturer's average profit $\Pi(\eta, \omega_p, t_p, P, Q_s)$.

Therefore, the corresponding optimization problem is as follows:

$$
\begin{aligned}
&\text{Maximize } \Pi(P, t_p, \eta, s, Q_s) \\
&\text{subject to } P > D(\omega_p, s),\ t_p > 0,\ s > 0,\ Q_s > 0,\ \eta > 0
\end{aligned} \tag{19}
$$

This is a highly nonlinear constrained maximization problem.

4 Solution methodology

Considering the second strategy (situation) taken by binary tournamenting process, three hybrid algorithms have been developed to solve the optimization problem (19). These algorithms are called as T2-AQPSO, T2-GQPSO and T2-AGQPSO. As the hybrid algorithms are depending on AQPSO, GQPSO and AGQPSO and also the tournamenting process, thus before discussing the hybrid algorithm, it is required to illustrate PSO, QPSO, AQPSO, GQPSO and AGQPSO algorithms and tournamenting process. The brief descriptions of these are given in the following subsections.

4.1 Particle Swarm optimization (PSO)

Particle swarm optimization is a prominent and efficient algorithm based on the observations of the social behavior of animals, such as fishes and birds. Here each solution of the swarm is considered as 'bird' or 'fish'-like volume free particle in activities. All the particles of the swarm fly throughout the search space aim to find the position of food (optimal position). At each iteration, particles in the swarm update their position by their personal experience and experience of the entire particles of the swarm. As a result, each particle has a memory to maintain its earlier best positions called 'personal best positions' with their own fitness. The position of the particle of the entire swarm which has highest fitness is called 'global best position'. Assume that such swarm of size N_p is moving in n_v-dimensional space. Let $u_i^K = (u_{i1}^K, u_{i2}^K, ..., u_{in_v}^K)$, $v_i^K = (v_{i1}^K, v_{i2}^K, ..., v_{in_v}^K)$, $p_i^K = (p_{i1}^K, p_{i2}^K, ..., p_{in_v}^K)$, $p_g^K = (p_{g1}^K, p_{g2}^K, ..., p_{gn_v}^K)$ be the current position, current velocity, personal best position and global best position, respectively, in the K-th iteration of the swarm. The velocity and position of particles are updated by the following rules:

$$
v_{ij}^{K+1} = v_{ij}^K + c_1 r_{1j}^K(p_{ij}^K - u_{ij}^K) + c_2 r_{2j}^K(p_{gj}^K - u_{ij}^K) \tag{20}
$$

$$
\begin{aligned}
&u_{ij}^{K+1} = u_{ij}^K + v_{ij}^{K+1} \\
&\text{for } i = 1, 2, ..., N_p; \\
&j = 1, 2, ..., n_v;\ K = 1, 2, ..., Max\_IT
\end{aligned} \tag{21}
$$

where $c_1 > 0$, $c_2 > 0$ are two constants, termed as the acceleration coefficients, and r_{ij}^K, r_{2j}^K are random numbers which follows uniform distribution in (0, 1).

4.2 Quantum-behaved PSO (QPSO)

Sometimes, traditional PSO algorithm is trapped by local optimum value and therefore cannot reach the global optimum position. To overcome these difficulties, Sun et al. (2005) proposed quantum-behaved PSO based on the quantum behavior of the particles. In quantum space, Newton's laws of motion is totally invalid because the position and velocity cannot be determined simultaneously according to Heisenberg's uncertainty principle. Hence, PSO algorithm is needed to design in terms of wave function model.

While a particle of mass M is moving in quantum space, the wave function $\psi(u, t)$ satisfies the Schrodinger wave equation in δ-potential well,

$$
\frac{d^2\psi}{du^2} + \frac{2M}{\hbar^2}[E + \gamma_1\delta(u - p)]\psi = 0 \tag{22}
$$

where E is the total energy of the particle; p is the center of potential of the particle; h and $\hbar = \frac{h}{2\pi}$ are the Planck's

constant and modified Planck's constant; and γ_1 is a positive constant which is chosen in such a way that it is always proportional to depth of the potential well.
Let $v = u - p$ then (22) reduces to,

$$\frac{d^2\psi}{dv^2} + \frac{2M}{\hbar^2}[E + \gamma_1\delta(v)]\psi = 0 \tag{23}$$

Thus, from (23) normalized wave function can be represented as

$$\psi(v) = \frac{1}{\sqrt{L}}e^{-\frac{|v-p|}{L}} \tag{24}$$

Thus, probability density function of the wave function is calculated as

$$Q = |\psi(v)|^2 = \frac{1}{L}e^{-\frac{2|v-p|}{L}} \tag{25}$$

From this equation, the probability of any particle appears at a certain position relative to p is found. But to get the fitness value, it is needed to determine the exact position of the particles. Therefore, to get the exact position, quantum state of the particle is to be collapsed into classical state. Monte Carlo simulation is used to measure this.
Monte Carlo simulation Since the value $\frac{1}{L}e^{-\frac{2|v-p|}{L}}$ always lies in the interval $\left(0, \frac{1}{L}\right)$, let us consider a random number in $\left(0, \frac{1}{L}\right)$ as $\frac{\xi}{L}$, where ξ is a random number in (0, 1). Now replace the value $\frac{\xi}{L}$ in place of Q in (25),

$$\frac{\xi}{L} = \frac{1}{L}e^{-\frac{2|v-p|}{L}} \tag{26}$$

$$\text{Hence, } v = \pm\frac{L}{2}log(\frac{1}{\xi}) \tag{27}$$

Thus, the equation (27) is going to be equal to

$$u = p \pm \frac{L}{2}log(\frac{1}{\xi}) \tag{28}$$

In the analysis of PSO, it is proved that p is the local attractor Γ of the particles. Hence, the equation (28) reduces to

$$u = \Gamma \pm \frac{L}{2}log(\frac{1}{\xi}) \tag{29}$$

Thus, in $K-$th iteration $j-$th component of $i-$th particle is updated as follows

$$u_{ij}^{K+1} = \Gamma_{ij}^K \pm \frac{L_{ij}^K}{2}log(\frac{1}{\xi_{ij}^{K+1}}) \tag{30}$$

The value of L_{ij}^K is calculated as

$$2\beta|u_{ij}^K - m_j^K| \tag{31}$$

where β is the contraction parameter (which plays an important role to control the convergence speed of the algorithm) and m_j^K is the mean best position defined by averages of the pbest positions of all the particles and

$$\begin{aligned}\text{Now, } m^K &= (m_1^K, m_2^K, ..., m_{n_v}^K)\\ &= \left(\frac{1}{N_p}\sum_{i=1}^{N_p} p_{i1}^K, \frac{1}{N_p}\sum_{i=1}^{N_p} p_{i2}^K, \ldots, \frac{1}{N_p}\sum_{i=1}^{N_p} p_{in_v}^K\right)\end{aligned} \tag{32}$$

also Γ_{ij}^K is defined as

$$\Gamma_{ij}^K = \phi_j p_{ij}^K + (1 - \phi_j)p_{gj}^K \tag{33}$$

It is proved in Sun et al. (2005) if $rand \geq 0.5$ then updating formula is

$$u_{ij}^{K+1} = \Gamma_{ij}^K + \frac{L_{ij}^K}{2}log\left(\frac{1}{\xi_{ij}^{K+1}}\right) \tag{34}$$

and if $rand < 0.5$ then updating formula is

$$u_{ij}^{K+1} = \Gamma_{ij}^K - \frac{L_{ij}^K}{2}log\left(\frac{1}{\xi_{ij}^{K+1}}\right) \tag{35}$$

Thus, the updating formula for QPSO algorithm is as follows

$$u_{ij}^{K+1} = \Gamma_{ij}^K + \beta|u_{ij}^K - m_j^K|log\left(\frac{1}{\varsigma_{ij}^{K+1}}\right), \text{if r} \geq 0.5 \tag{36}$$

$$u_{ij}^{K+1} = \Gamma_{ij}^K - \beta|u_{ij}^K - m_j^K|log\left(\frac{1}{\varsigma_{ij}^{K+1}}\right), \text{if r} < 0.5 \tag{37}$$

where ξ_{ij}^{K+1} is a random number in (0, 1), $\phi_j \sim U(0, 1)$ and r is the random number in (0, 1).

4.3 Gaussian quantum-behaved PSO (GQPSO)

To avoid the premature convergence Coelho (2010) proposed GQPSO algorithm. In this version of PSO, particles of the swarm are more volatile and diversify. Here, the QPSO attractor $\Gamma_{ij}^K = \phi_j p_{ij}^K + (1 - \phi_j)p_{gj}^K$ is replaced by $\Gamma_{ij}^K = \frac{G_1 p_{ij}^K + G_2 p_{gj}^K}{G_1 + G_2}$, $j = 1, 2, ..., n_v$ and the random number ς_{ij}^{K+1} is replaced by the Gaussian random numbers G_{ij}^{K+1} in the QPSO algorithm, where G_1 and G_2 are the random numbers generated by Gaussian probability distribution with zero mean and unit variance.

4.4 Adaptive quantum-behaved PSO (AQPSO)

In AQPSO (Xu and Sun 2005), the analysis of control parameters is studied in detail, which was not discussed in primary QPSO so far. Here, the parameter β (known as creativity coefficient) plays an important role and it is dependent on attraction and repulsion phases. When the particles are in attraction phase, the diversity of the particles increases and in this situation one has to assign $\beta = \beta_a$, where $\beta_a \leq 1$ as far as in repulsion phase one has to $\beta = \beta_r$, where $\beta_r > 1$. Diversity of the swarm at K-th iteration is defined as

$$d = \frac{1}{N_p|L|}\sum_{i=1}^{N_p}\sqrt{\sum_{j=1}^{n_v}(p_{ij}^K - m_j^K)^2} \tag{38}$$

where $|L|$ is the longest diagonal of the search space.

The position of the particles at the K-th iteration is updated by using the following rules

$$u_{ij}^{K+1} = \Gamma_{ij}^K + \beta|u_{ij}^K - m_j^K|log\left(\frac{1}{\varsigma_{ij}^{K+1}}\right), \text{if r} \geq 0.5 \tag{39}$$

$$u_{ij}^{K+1} = \Gamma_{ij}^K - \beta|u_{ij}^K - m_j^K|log\left(\frac{1}{\varsigma_{ij}^{K+1}}\right), \text{if r} < 0.5 \tag{40}$$

where β runs through the β_r to β_a of the swarm and if $d < dlow$ assign $\beta = \beta_a$ and if $d > dhigh$ assign $\beta = \beta_r$.

4.5 Adaptive Gaussian quantum-behaved PSO (AGQPSO)

On inspiring to get the advantages of AQPSO and GQPSO algorithms, Kumar et al. (2019) proposed AGQPSO algorithm. Here, to avoid the premature convergence, Gaussian attractor and Gaussian random number have been used. Over and above, to make less susceptible to stuck by stagnation fault, parameter controls have been made. Thus, proposed AGQPSO technique can be called as a modification of AQPSO technique. In AQPSO technique, QPSO attractor $\Gamma_{ij}^K = \phi_j p_{ij}^K + (1-\phi_j)p_{gj}^K$ has been used for compelling the particles towards the global optimum position. But in AGQPSO, Gaussian attractor $\Gamma_{ij}^K = \frac{G_1 p_{ij}^K + G_2 p_{gj}^K}{G_1+G_2}$, $j = 1, 2, ..., n_v$ has been used in place of the QPSO attractor $\Gamma_{ij}^K = \phi_j p_{ij}^K + (1-\phi_j)p_{gj}^K$. Also, Gaussian random numbers G_{ij}^{K+1} have been used. The pseudo-code AGQPSO is as follows:

Pseudo-code

```
begin
    initialize AGQPSO parameters and bounds of decision variables
    generate a swarm of particles randomly
    compute the fitness of all particles
    store the initial position (generate randomly) of each particle and its fitness 'pbest'
    and 'pbestobj' respectively
    find the global best among all particles as 'gbest'
    while (termination criterion satisfied)
          calculate 'mbest' (mean best position)
          measure the diversity (d) of the swarm
          if d < dlow assign β = β_a
          if d > dhigh assign β = β_r
          generate Gaussian random numbers and calculate Gaussian
          attractor
          update the position of each particle
          find 'pbest'
          find 'gbest'
    end while
    print the best result
  end
```

4.6 Tournamenting QPSOs

In genetic algorithm, sometimes researchers have to run the program several times by taking different populations in different runs to get the best solutions from them. For smaller search space, this process gives better outputs but for broad solution space, and for non-convex/non-concave problem, it becomes arduous. To overcome this type of difficulties, Bhunia and Samanta (2014) proposed an algorithm, known as tournament genetic algorithm as an alternative technique.

4.6.1 Hybrid binary tournament

In computational optimization, to make more efficient than original algorithm/algorithms, hybrid algorithms are proposed. Hybrid algorithm refers to the combination of two or more algorithms or embedding an algorithm in terms of different fashion (like tournament fashion, league fashion, chaotic mapping fashion). As a result, the new formed hybrid algorithm is better than the original algorithm/algorithms.

Actually in any game to select the best team among all the teams is arranged through tournament. This tournament can be designed in different fashions, e.g., binary tournament, league tournament, etc. In binary tournament, in each game one team is selected out of two teams in every round of the tournament. The whole tournament is performed in different rounds dependent on the number of teams which take part in the game.

4.6.2 Hybrid binary tournament-based QPSOs

In this work, AQPSO, GQPSO and AGQPSO techniques have been applied to elevate the swarm of particles in each round of tournament process. Here four teams' tournamenting process has been considered. Firstly, two swarms S_1, S_2 have been updated into S'_1, S'_2 and finally taken 50% particles from S'_1, S'_2. This swarm is renamed as S'_{12}. 50% particles may be considered from the improved swarms S'_1 and S'_2 with the help of the following strategies:

Strategy-1: Alliance of the best 50% from each of the modified swarms S'_1, S'_2.
Strategy-2: Chosen of the best 50% from the alliance of two swarms S'_1, S'_2.
Strategy-3: Creation of a new swarm by choosing a better individual by making a comparison between randomly taken particles from each of the swarms S'_1, S'_2.
Strategy-4: Chosen of better swarm which carries the best particles between two swarms S'_1, S'_2.
Strategy-5: Chosen of better swarm which carries better average fitness value.
Strategy-6: Chosen of better swarm which carries finer standard deviation of the fitness values of the particles.

It should be noted that in a real game, the better one among two teams is selected for the next round. However, in simulation process, different strategies may occur for the computational optimization.

In the current work, using three rounds, all embedded/hybridized PSO algorithms (AQPSO, GQPSO and WQPSO) are designed in the form of second strategy-based tournament fashion using four teams. As a result, the said algorithms are hybridized in the form of binary tournament fashion. The details of binary tournament-based algorithms and their different scenarios are found in Kumar et al. (2019), Kumar et al. (2020), Akhtar et al. (2020).

4.7 Pseudo-code of binary tournamenting QPSOs

```
begin
    n ← 1
    while (n < 4) do
        initialize S_n where S_n denotes the n − th swarm
        apply AQPSO/GQPSO/AGQPSO on S_n to get the swarm S'_n
        initialize S_{n+1}
        apply AQPSO/GQPSO/AGQPSO on S_{n+1} to get the swarm S'_{n+1}
        find 50% particles from S'_n and S'_{n+1} as per any one of the strategies
        1 − 6 and obtain improved swarm S'_{n,n+1}
        apply AQPSO/GQPSO/AGQPSO on S'_{n,n+1} to get the swarm S''_{n,n+1}
        n ← n + 2
    end while
    find 50% particles from S''_12 and S''_34 as per Strategy-2 and obtain the swarm S''_1234
    apply AQPSO/GQPSO/AGQPSO on S''_1234 to get the swarm S'''_1234
    save the best found result
end
```

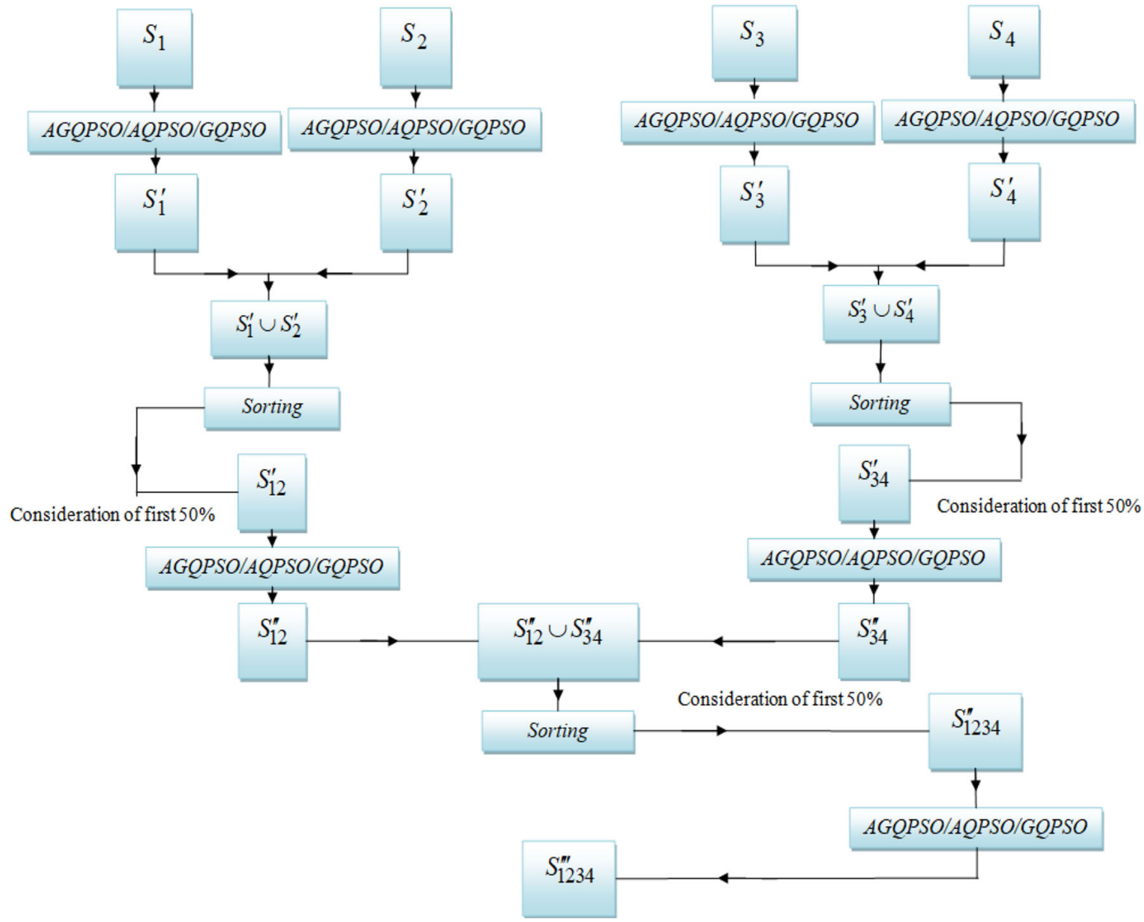

Fig. 5 Pictorial representation of second strategy of tournament (T2) with four teams

5 Numerical experiment

In this section, different types of numerical experiments have been considered. In subsect. 5.1, three numerical examples are considered to check the validation of the proposed model as well as the robustness of the hybrid algorithms. Here, the best found solutions, worst found solutions, statistical results and convergence history for each of the problems have been provided. In subsect. 5.2, the discussion of the obtained results is performed. The four different nonparametric statistical tests are performed in subsect. 5.3.

5.1 Numerical illustration

To validate the proposed model, three different numerical examples have been considered which are as follows.

Example 1: In this example the values of different parameters are taken as $C_0 = Rs.80$ per unit, $C_1 = Rs.0.01$ per unit, $C_2 = Rs.500$ per unit, $C_3 = Rs.2$ per unit, $C_4 = Rs.2.5$ per unit, $\alpha = 1.25 \in R^+$, $\beta_1 = 310$ unit, $\lambda_1 = 2.0$ unit, $\lambda_2 = 0.6$ unit, $\gamma = 0.5 \in R^+$, $\delta = 2 \in R^+$, $w_p = 0.5$ year, $\lambda = 0.3 \in R^+$, $\mu = 0.5 \in R^+$, $\theta_0 = 0.1 \in (0, 1)$, $\vartheta = 0.09 \in R^+$, $\xi = 0.5 \in R^+$, $h = Rs.6.0$ per unit/ unit time, $a = Rs.20.0$ per unit, $b = Rs.5$ per unit, $c_b = Rs.20.0$ per unit, $A = Rs.300$ per cycle.

Example 2: In this example the values of different parameters are taken as $C_0 = Rs.80$ per unit, $C_1 = Rs.0.03$ per unit, $C_2 = Rs.490$ per unit, $C_3 = Rs.2.2$ per unit, $C_4 = Rs.2.9$ per unit, $\alpha = 1.15 \in R^+$, $\beta_1 = 300$ unit, $\lambda_1 = 1.5$ unit, $\lambda_2 = 0.5$ unit, $\gamma = 0.4 \in R^+$, $\delta = 4.2 \in R^+$, $w_p = 0.7$ year, $\lambda = 0.6 \in R^+$, $\mu = 0.4 \in R^+$, $\theta_0 = 0.2 \in (0, 1)$, $\vartheta = 0.04 \in R^+$, $\xi = 1.5 \in R^+$, $h = Rs.6.0$ per unit/ unit time, $a = Rs.14.0$ per unit, $b = Rs.14.0$ per unit, $c_b = Rs.5.5$ per unit, $A = Rs.305$ per unit.

Example 3: In this example the values of different parameters are taken as $C_0 = Rs.82$ per unit, $C_1 = Rs.0.02$ per unit, $C_2 = Rs.510$ per unit, $C_3 = Rs.1.5$ per unit, $C_4 = Rs.2.3$ per unit, $\alpha = 1.10 \in R^+$, $\beta_1 = 305$ unit, $\lambda_1 = 2.4$ unit, $\lambda_2 = 0.8$ unit, $\gamma = 0.6 \in R^+$, $\delta = 3.5 \in R^+$, $w_p = 0.6$ year, $\lambda = 0.2 \in R^+$, $\mu = 0.8 \in R^+$, $\theta_0 = 0.15 \in (0, 1)$, $\vartheta = 0.1 \in R^+$, $\xi = 0.9 \in R^+$, $h = Rs.5.0$ per unit/ unit time, $a = Rs.16.0$ per unit, $b = Rs.4.0$ per unit, $c_b = Rs.16.0$ per unit, $A = Rs.300$ per unit.

Table 2 Parameters of different algorithms

Parameters	Algorithms					
	T2-AGQPSO	T2-AQPSO	T2-GQPSO	AGQPSO	AQPSO	GQPSO
N_p	15	15	15	25	25	25
Max_gen	30	30	30	130	130	130

The best found average profit has been obtained using six different algorithms including hybrid tournamenting algorithms. Since all the algorithms are probabilistic in nature, so 30 independent runs have been performed for each of the algorithms. Each algorithm has been coded by using C++ software, and all runs have been done on a Laptop core i3-7020U CPU, 7th generations, 2.30 GHz processors in LINUX environment. Simulation parameters of different algorithms are provided in Table 2.

Here, two different types of swarm size and maximum generations have been considered in two types of algorithms. Since for small swarm size and generations, functions evaluations are large in binary tournament-based algorithms, to compare these two types of algorithms, the swarm size and maximum generations have to set in such a way that difference of function evaluations in two types of algorithms is minimum. On considering swarm size 15 and maximum generation 30 in tournament-based PSOs, the total number of function evaluations is $7 \times 15 \times 30 + 4 \times 15 = 3210$, while in general PSOs the total number of function evaluations is $25 \times 130 + 25 = 3275$. Therefore, the difference of function evaluations is not so large. However, function evaluations are small in tournament-based PSOs. Besides, other parameters of PSOs are considered as per the directions of Coelho (2010) and Sun et al. (2005). To test the efficiency of the hybrid tournament PSOs, best found results, worst results and statistical measurements of the average profit are shown in Tables 3, 4, 5, 6, 7, 8, 9, 10, 11 for each of the problems.

From the results, it is observed that the hybrid tournament PSOs give better results than its general algorithm with minimum function evaluations and CPU times for each problem.

5.2 Results discussion

The best found solutions of Example 1 are the same for all the algorithms. However, the worst found solutions are different for all the algorithms. The difference of best found solutions and worst found solutions is minimum for T2-AQPSO algorithms. Also from Tables 3, 4 and 5, it is observed that T2-AQPSO algorithm performs better with respect to best found objective value, worst objective value, mean objective value and standard deviation of the objective values with minimum CPU times as well as function evaluations. Though AGQPSO algorithm does not perform so well, it has satisfactory performance in tournament hybridization.

The best found solutions of Example 2 be the same for all the algorithms except for T2-AGQPSO. However, it is same up to eight decimal places. Here also, the difference between the found solutions and worst found solutions is minimum for T2-AQPSO algorithms. Here T2-AQPSO algorithm performs better with respect to the best objective value, worst objective value and standard deviation. Here, mean and median of the objective values are better for AQPSO algorithm, but it has larger CPU times and function evaluations. So, overall T2-AQPSO algorithm performs better for Example 2 also.

In discussions of Example 3, it is observed from Tables 9, 10 and 11 that the best found solutions be the same for all the algorithms. The differences between best found solutions and worst found solutions are very less for T2-AQPSO and AQPSO algorithms. From the statistical data, it is observed that their performances be the same except standard deviations. The standard deviation is less for T2-AQPSO algorithm. Also, it should be noted that T2-AQPSO algorithm produces the best found result with minimum CPU times and function evaluations. Here also, AGQPSO and its tournament hybridized version perform well than other hybrid algorithms.

So in overall discussions, it does not have any hesitance to say that T2-AQPSO is the best algorithm than others in performance for this profit optimization problem. Of course, it should be noted that the computational results obtained in Tables Tables 9, 10 and 11 do not establish that T2- AQPSO algorithm performs well for all types of optimization problems (Wolpert and Macready 1997). Hence, this hybrid T2-AQPSO algorithm has the highest performance for the optimization problem (19) only.

The convergence history of the best found solutions obtained from AGQPSO, AQPSO and GQPSO algorithms is shown in Figures 6, 7 and 8. However, the convergence graphs of different hybrid algorithms cannot be drawn as the parent metaheuristic algorithms like AQPSO, AGQPSO and GQPSO are applied in different rounds of the tournament. From the convergence history, it is observed that AQPSO performs better for Example 1 and Example 3, whereas AGQPSO performs better for Example 2.

Table 3 Best found results of Example 1

Variables/Objective	Obtained best found values by different algorithms					
	T2-AGQPSO	T2-AQPSO	T2-GQPSO	AGQPSO	GQPSO	AQPSO
P^* (unit)	1243.929953	1243.929755	1243.929854	1243.929984	1243.930072	1243.930084
t_p^* (year)	0.295506	0.295506	0.295506	0.295506	0.295506	0.295506
η^* ($\in R^+$)	0.807478	0.807478	0.807478	0.807478	0.807478	0.807477
Q_s^* (unit)	7.938465	7.938451	7.93846	7.938452	7.938464	7.938468
s^* (Rs.)	314.532238	314.532238	314.532236	314.532239	314.532234	314.53223
T_1^* (year)	2.805355	2.805355	2.805356	2.805355	2.805355	2.805355
T_2^* (year)	2.873275	2.873275	2.873275	2.873275	2.873275	2.873275
T^* (year)	2.880915	2.880914	2.880915	2.880915	2.880915	2.880915
$\Pi(P^*, t_p^*, \eta^*, s^*, Q_s^*)$ (Rs.)	23866.088717850	23866.088717850	23866.088717850	23866.088717850	23866.088717850	23866.088717850
CPU time (second)	0.017534	0.013774	0.017363	0.017805	0.013956	0.017592

Table 4 Worst found results of Example 1

Variables/Objective	Obtained best found values by different algorithms					
	T2-AGQPSO	T2-AQPSO	T2-GQPSO	AGQPSO	GQPSO	AQPSO
P (unit)	1243.966617	1243.950285	1203.434669	1238.42655	1243.892416	1242.690296
t_p (year)	0.295495	0.295500	0.301622	0.296906	0.295516	0.295733
η ($\in R^+$)	0.807476	0.807479	0.803694	0.806027	0.807479	0.807093
Q_s (unit)	7.93861	7.938568	7.611321	7.881211	7.937135	7.942437
s (Rs.)	314.531864	314.53211	314.58223	314.994503	314.532835	314.531725
T_1 (year)	2.805331	2.805345	2.770498	2.812409	2.805377	2.804665
T_2 (year)	2.873252	2.873265	2.835523	2.879978	2.873285	2.872621
T (year)	2.880892	2.880905	2.843125	2.887598	2.880924	2.880273
$\Pi(P, t_p, \eta, s, Q_s)$ (Rs.)	23866.088717071	23866.088717635	23865.570527293	23865.947160283	23866.088716444	23866.088154303
CPU time (second)	0.018458	0.013792	0.017325	0.017806	0.014017	0.017654

Table 5 Statistical results of Example 1 using different algorithms

Objective	Algorithms	Statistical Measurement			Average CPU time (Sec.)	Average function evaluation
		Mean	Median	Standard deviation		
	T2-AGQPSO	23866.088717759	23866.088717838	1.79394×10^{-07}	0.017756333	3210
Average profit	T2-AQPSO	23866.088717823	23866.088717840	4.30315×10^{-08}	0.013853433	3210
of manufacturer	T2-GQPSO	23866.049913249	23866.088716212	0.118401651	0.017359033	3210
$\Pi(P^*, t_p^*, \eta^*, s^*, Q_s^*)$	AGQPSO	23866.082351747	23866.088706698	0.025848005	0.017836333	3275
	AQPSO	23866.088717803	23866.088717850	2.56609×10^{-07}	0.014099367	3275
	GQPSO	23866.088698067	23866.088717848	1.02772×10^{-04}	0.0176369	3275

Table 6 Best found results of Example 2

Variables/Objective	Obtained best found values by different algorithms					
	T2-AGQPSO	T2-AQPSO	T2-GQPSO	AGQPSO	GQPSO	AQPSO
P^* (unit)	2419.893929	2419.896361	2419.895447	2419.895778	2419.89656	2419.896092
t_p^* (year)	0.139453	0.139453	0.139453	0.139453	0.139453	0.139453
η^* ($\in R^+$)	0.948031	0.94803	0.94803	0.94803	0.94803	0.94803
Q_s^* (unit)	6.763167	6.763178	6.763156	6.763176	6.763177	6.763178
s^* (Rs.)	361.926752	361.92675	361.926728	361.926753	361.926746	361.926748
T_1^* (year)	2.674587	2.674587	2.674587	2.674588	2.674588	2.674588
T_2^* (year)	2.732869	2.73287	2.732869	2.732871	2.73287	2.73287
T^* (year)	2.735967	2.735967	2.735967	2.735968	2.735968	2.735968
$\Pi(P^*, t_p^*, \eta^*, s^*, Q_s^*)$ (Rs.)	27857.755751528	27857.755751529	27857.755751529	27857.755751529	27857.755751529	27857.755751529
CPU time (second)	0.017836	0.014034	0.017565	0.018728	0.014227	0.017934

Table 7 Worst found results of Example 2

Variables/Objective	Obtained best found values by different algorithms					
	T2-AGQPSO	T2-AQPSO	T2-GQPSO	AGQPSO	GQPSO	AQPSO
P (unit)	2419.859214	2419.865588	2407.733974	2417.586579	2419.917524	2417.998015
t_p (year)	0.13945	0.139454	0.140099	0.139702	0.139453	0.139578
η ($\in R^+$)	0.94801	0.948031	0.953151	0.949271	0.94803	0.947889
Q_s (unit)	6.76197	6.763272	6.758161	6.705255	6.76274	6.760435
s (Rs.)	361.929366	361.926826	361.90648	361.881998	361.92677	361.935936
T_1 (year)	2.67451	2.674574	2.67428	2.676547	2.674609	2.674942
T_2 (year)	2.732783	2.732857	2.732512	2.734302	2.732887	2.733202
T (year)	2.73588	2.735955	2.735623	2.737376	2.735985	2.736301
$\Pi(P, t_p, \eta, s, Q_s)$ (Rs.)	27857.755745890	27857.755751436	27857.678865436	27857.747951525	27857.755751377	27857.755173185
CPU time (second)	0.017881	0.014041	0.017594	0.01873	0.014283	0.017952

Table 8 Statistical results of Example 2 using different algorithms

Objective	Algorithms	Statistical Measurement			Average	Average
		Mean	Median	Standard deviation	CPU time (Sec.)	function evaluation
	T2-AGQPSO	27857.755751258	27857.755751509	1.01966×10^{-06}	0.0180197	3210
Average profit	T2-AQPSO	27857.755751518	27857.755751525	1.89362×10^{-08}	0.014045967	3210
of manufacturer	T2-GQPSO	27857.751669065	27857.755747243	0.014775741	0.017593867	3210
$\Pi(P^*, t_p^*, \eta^*, s^*, Q_s^*)$	AGQPSO	27857.755347056	27857.755750614	0.001500409	0.018701467	3275
	AQPSO	27857.755751520	27857.755751529	3.03187×10^{-08}	0.014257233	3275
	GQPSO	27857.755731481	27857.755751528	1.05468×10^{-04}	0.0179352	3275

Table 9 Best found results of Example 3

Variables/Objective	Obtained best found values by different algorithms					
	T2-AGQPSO	T2-AQPSO	T2-GQPSO	AGQPSO	GQPSO	AQPSO
P^* (unit)	326.194509	326.19453	326.194481	326.194416	326.194323	326.194361
t_p^* (year)	0.804573	0.804574	0.804574	0.804574	0.804574	0.804574
η^* ($\in R^+$)	0.988174	0.988174	0.988174	0.988174	0.988174	0.988174
Q_s^* (unit)	5.890809	5.890809	5.890806	5.890803	5.890803	5.890803
s^* (Rs.)	240.173012	240.173014	240.173013	240.173009	240.173011	240.173013
T_1^* (year)	2.154569	2.15457	2.15457	2.15457	2.15457	2.15457
T_2^* (year)	2.208162	2.208163	2.208163	2.208163	2.208163	2.208163
T^* (year)	2.238688	2.238689	2.238688	2.238689	2.238689	2.238689
$\Pi(P^*, t_p^*, \eta^*, s^*, Q_s^*)$ (Rs.)	15697.010666910	15697.010666910	15697.010666910	15697.010666910	15697.010666910	15697.010666910
CPU time (second)	0.017947	0.014053	0.017596	0.018077	0.01427	0.017922

Table 10 Worst found results of Example 3

Variables/Objective	Obtained best found values by different algorithms					
	T2-AGQPSO	T2-AQPSO	T2-GQPSO	AGQPSO	GQPSO	AQPSO
P (unit)	326.206268	326.193221	326.24746	326.202142	326.195001	326.151212
t_p (year)	0.804548	0.804578	0.804445	0.804548	0.804574	0.804722
η ($\in R^+$)	0.988171	0.988174	0.988166	0.988168	0.988174	0.9882
Q_s (unit)	5.890899	5.89077	5.891605	5.891458	5.8908	5.888125
s (Rs.)	240.172879	240.172999	240.17344	240.172988	240.172955	240.172575
T_1 (year)	2.154578	2.154572	2.15458	2.154549	2.154573	2.1546775
T_2 (year)	2.208171	2.208165	2.20818	2.208148	2.208165	2.208244
T (year)	2.238696	2.238691	2.238702	2.238676	2.238691	2.238762
$\Pi(P, t_p, \eta, s, Q_s)$ (year)	15697.010666615	15697.010666906	15697.010661282	15697.010666573	15697.010666906	15697.010659238
CPU time (second)	0.017974	0.014163	0.01763	0.018125	0.014266	0.013872

Table 11 Statistical results of Example 3 using different algorithms

Objective	Algorithms	Statistical Measurement				
		Mean	Median	Standard deviation	Average CPU time (Sec.)	Average function evaluation
	T2-AGQPSO	15697.010666900	15697.010666910	5.38081×10^{-08}	0.017939933	3210
Average profit	T2-AQPSO	15697.010666910	15697.010666910	9.22467×10^{-10}	0.014095933	3210
of manufacturer	T2-GQPSO	15697.010666549	15697.010666909	1.140061×10^{-06}	0.017608567	3210
$\Pi(P^*, t_p^*, \eta^*, s^*, Q_s^*)$	AGQPSO	15697.010666872	15697.010666906	8.77108×10^{-08}	0.018087933	3275
	AQPSO	15697.010666910	15697.010666910	1.01484×10^{-09}	0.014322433	3275
	GQPSO	15697.010666345	15697.010666910	1.88612×10^{-06}	0.017669	3275

5.3 Nonparametric statistical test

From the earlier mentioned statistical results, it is observed that T2-AQPSO algorithm performs better than all other algorithms (T2-AGQPSO, T2-GQPSO, AGQPSO, AQPSO and GQPSO) in overall comparison. But it is needed to perform some statistical tests due to randomness of the metaheuristic algorithm to check the significance of the results. In this section, four nonparametric statistical tests, viz. Wilcoxon rank-sum test (Derrac et al. 2011; Duary et al. 2020; García et al. 2009), Friedman test (Derrac et al. 2011), Wilcoxon signed rank test (García et al. 2009) and Iman and Davenport test (Derrac et al. 2011), have been performed on the 30 objective values (average profit values) obtained from 30 runs to compare with other used algorithms for each of the examples. By considering T2-AQPSO as the control algorithm, the p values of the different methods are presented in Tables 12, 13, 14. All p values are calculated by using Microsoft Office excel 2007.

From Table 12, it is seen that the superiorities of T2-AQPSO algorithm are statistically significant for T2-GQPSO, AGQPSO and AQPSO algorithms for Example 1 as the p values are less than 0.05. However, T2-AGQPSO and GQPSO perform very similar with T2-AQPSO algorithm.

Also from Table 13, it is noticed that the superiorities of T2-AQPSO algorithm are statistically significant for T2-AGQPSO,T2-GQPSO, AGQPSO and AQPSO algorithms for Example 2 as the p values are less than 0.05. However, the performance of GQPSO algorithm is very comparative with T2-AQPSO algorithm.

In Table 14, it is seen that the superiorities of T2-AQPSO algorithm are statistically significant for T2-GQPSO and AGQPSO algorithms for Example 3 at 5% level of significance. Here, the performance of T2-AGQPSO, AQPSO and GQPSO algorithms is very similar with T2-AQPSO algorithm.

6 Sensitivity analysis

For the proposed model, the sensitivity analyses by changing the values of known important model parameters A, c_b, C_0, β_1, θ_0, w_p, h and α from -20% to 20% have been demonstrated. Also, the effects of P, t_p, s, η, Q_s, T_1, T_2, T, Π with respect to each parameter have been analyzed and the results are shown graphically in Figs. 9, 10, 11, 12, 13, 14, 15 and 16.

The effectiveness of the parameters A, c_b, C_0, β_1, θ_0, w_p, h and α on the best found solution of maximization problem (17) (P, t_p, s, η, Q_s, T_1, T_2, T, Π) is measured by the following scales:

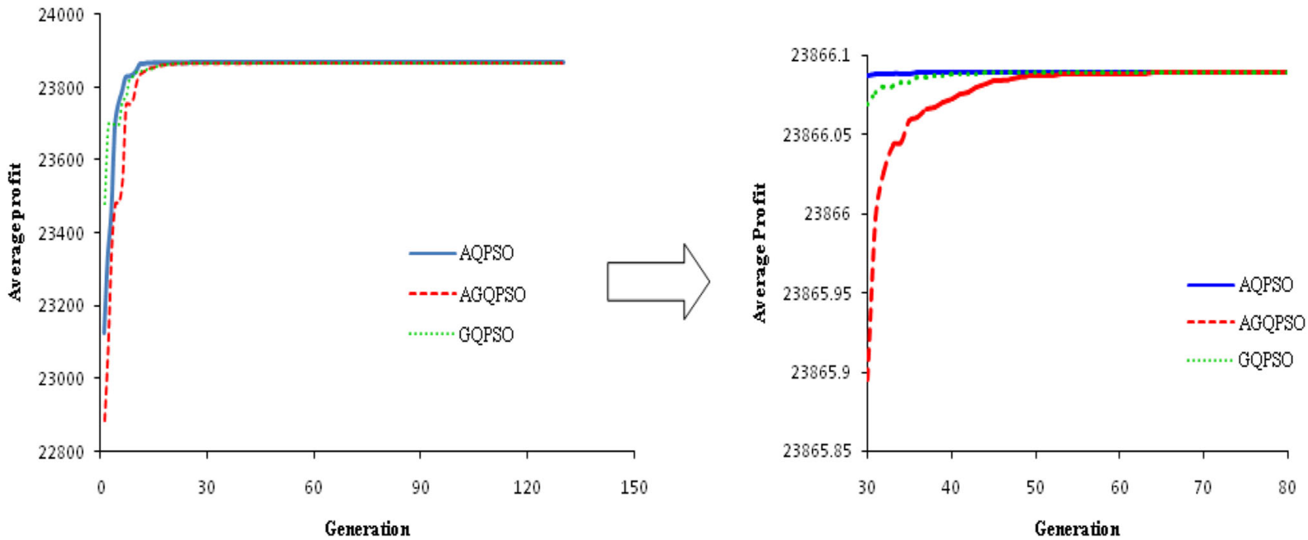

Fig. 6 Convergence history of the best found solution of different algorithms for Example 1

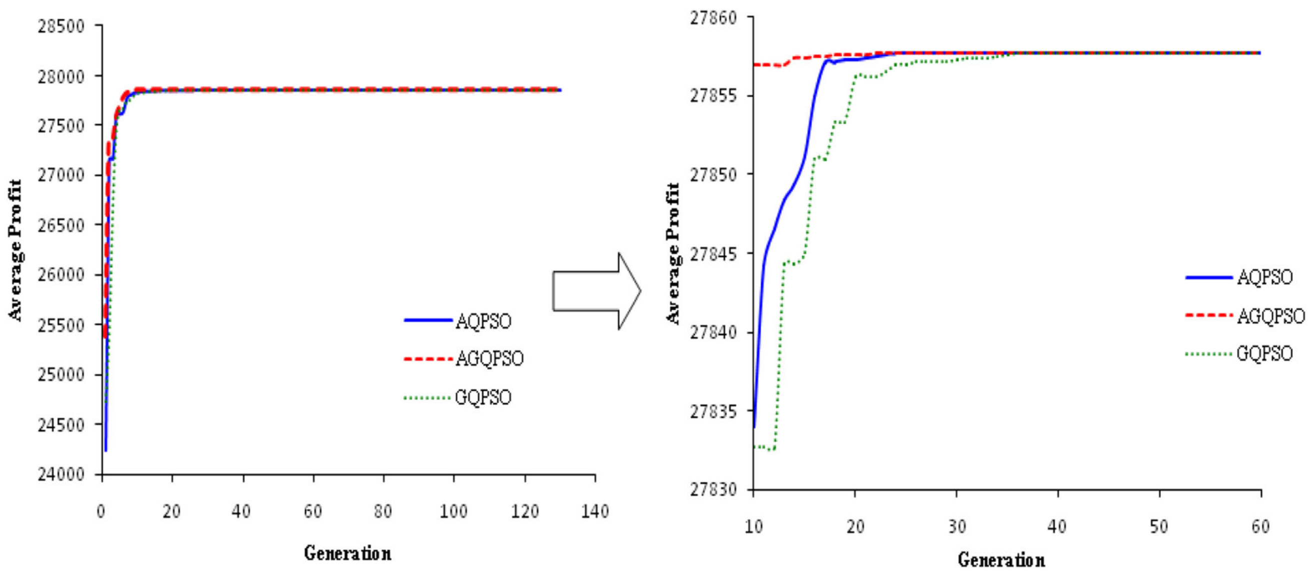

Fig. 7 Convergence history of the best found solution of different algorithms for Example 2

(i) The model parameters are known as highly sensitive directly (reversely) with respect to the best found solution of maximization problem (19) if the said solutions change from less than -20% (greater than +20%) to greater than +20% (less than -20%) with the changes of that parameters from -20% to +20%.
(ii) The model parameters are known as equally sensitive with directly (reversely) with respect to the best found solution of maximization problem (19) if the said solutions change from -20% (+20%) to +20% (-20%) with the changes of that parameters from -20% to +20%.
(iii) The model parameters are known as moderately sensitive directly (reversely) with respect to the best found solution of maximization problem (19) if the said solutions change from near -10% (+10%) to near +10% (-10%) with the changes of that parameters from -20% to +20%.
(iv) The model parameters are known as less sensitive directly (reversely) with respect to the best found solution of maximization problem (19) if the said solutions change from greater than -10% (less than +10%) to less than +10% (greater than-10%) with the changes of that parameters from -20% to +20%.
(v) The model parameters are known as insensitive with respect to the best found solution of maximization problem (19) if the said solution changes from greater than -1% (+1%) to less than +1% (-1%) with the changes of that parameters from -20% to +20%.

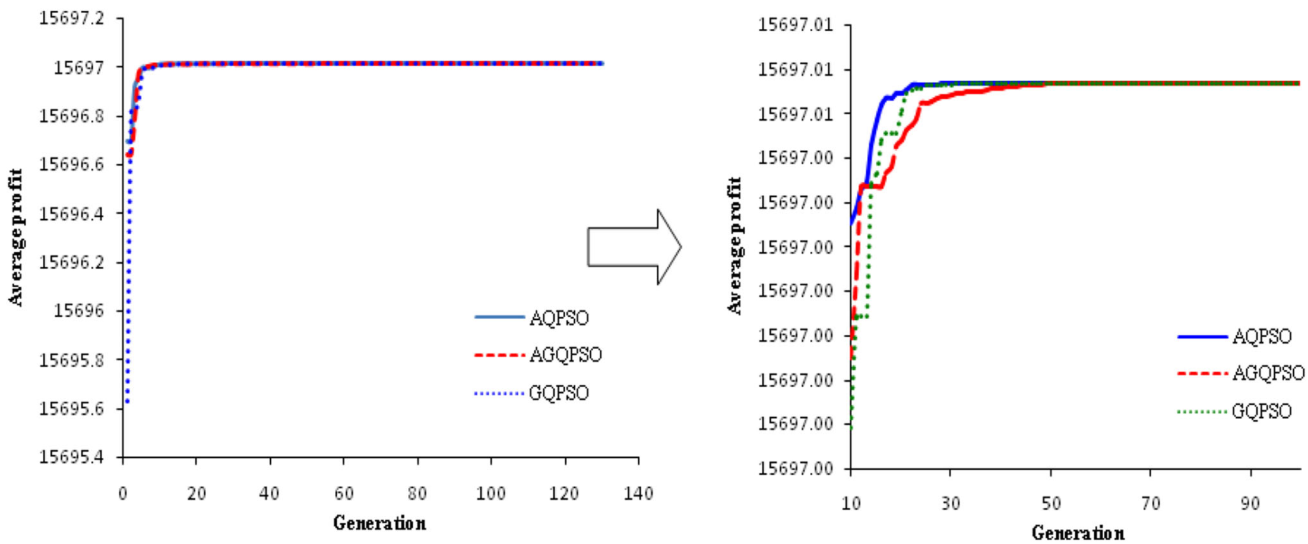

Fig. 8 Convergence history of the best found solution of different algorithms for Example 3

Table 12 Nonparametric statistical tests of Example 1

T2-AQPSO vs.	p value			
	Wilcoxon rank sum test	Friedman test	Iman and Davenport test	Wilcoxon signed rank sum test
T2-AGQPSO	0.728265296	0.583882423	0.592483283	0.217727087
T2-GQPSO	4.26655×10^{-06}	0.00026073	4.22213×10^{-05}	4.44934×10^{-05}
AGQPSO	7.08811×10^{-08}	5.90358×10^{-05}	2.69579×10^{-06}	8.89867×10^{-05}
AQPSO	1.41762×10^{-07}	5.01033×10^{-06}	5.94311×10^{-09}	9.88267×10^{-05}
GQPSO	0.111986872	0.361310432	0.37019003	0.665349081

Table 13 Nonparametric statistical tests of Example 2

T2-AQPSO vs.	p value			
	Wilcoxon rank sum test	Friedman test	Iman and Davenport test	Wilcoxon signed rank sum test
T2-AGQPSO	0.001236185	0.0061699	0.004180638	0.000561614
T2-GQPSO	1.86085×10^{-06}	0.000126046	1.16972×10^{-05}	7.22961×10^{-06}
AGQPSO	2.37682×10^{-07}	0.000126046	1.16972×10^{-05}	7.22516×10^{-06}
AQPSO	0.001335591	0.028459739	0.025774236	0.034417232
GQPSO	0.813005288	1.0	1.0	0.061883751

Table 14 Nonparametric statistical tests of Example 3

T2-AQPSO vs.	p value			
	Wilcoxon rank sum test	Friedman test	Iman and Davenport test	Wilcoxon signed rank sum test
T2-AGQPSO	0.813005288	0.855132141	0.858711513	0.714431128
T2-GQPSO	0.001596887	0.001910775	0.000888826	0.000292478
AGQPSO	0.000571543	0.0061699	0.004180638	0.00026984
AQPSO	0.544407554	0.583882423	0.592483283	0.731600589
GQPSO	0.078518652	0.067889173	0.066882888	0.004048316

From Figs. 9, 10, 11, 12, 13, 14, 15 and 16, the following observations can be made:

(i) From Fig. 9, it is observed that P, t_p, Q_s, T_1, T_2 are less sensitive directly against the changes of setup cost (A). On the other hand, s, η, T, Π are insensitive with respect to A.
(ii) Figure 10 shows that Q_s is less sensitive directly with respect to backordering cost (c_b). On the other hand, $P, t_p, \eta, s, \eta, T_1, T, \Pi$ are insensitive with the changes of c_b.
(iii) From Fig. 11, it is seen that s, Q_s, T_1, T_2, T are less sensitive directly against the changes of fixed production cost (C_0). Also η has moderate impact reversely with respect to C_0. On the other hand, P, t_p are slightly reverse sensitive and average profit (π) is moderately reverse sensitive with the changes of C_0.
(iv) In view of Fig. 12, it is clear that P, t_p, s are moderately sensitive directly and average profit (Π) has direct impact largely with the changes of fixed demand (β_1). Furthermore, Q_s, T_1, T_2, T have reverse sensitive moderately and η is insensitive with the changes of β_1.
(v) In Fig. 13, it is obvious that P, Q_s are less sensitive directly and t_p, π are reverse sensitive slightly with the changes of defective rate parameter θ_0. Also η is largely sensitive directly and s, T_1, T_2, T are insensitive with the changes of θ_0.
(vi) Figure 14 exposes that t_p has less sensitive directly and P has less sensitive reversely with the change of warranty period (w_p). On the other hand, s, η, Q_s, T_1, T_2, T, Π are insensitive with the changes of w_p.
(vii) From Fig. 15, it is observed that P, T_1 and T are equally reverse sensitive and t_p is reverse sensitive moderately with the changes of holding cost (h). Also Q_s is moderately sensitive directly and Π is reverse sensitive slightly with the changes of h. On the other hand, s, η, T_2 are insensitive with the changes of h.
(viii) Finally, Fig. 16 exposes that Q_s has largely sensitive reversely and $s, \eta, Q_s, T_1, T_2, T, \Pi$ are insensitive with the backlogged parameter α.

7 Managerial insights

The proposed production inventory model can be applied to any manufacturing systems where the production system produces perfect as well as defective items. The manager of any manufacturing company will invest development cost to reduce the production of defective items. The management of manufacturing company gives the product warranty to increase annual customers' demand and profit of the company. On the other hand, the customers will be benefited due to the product warranty. Here, the customers' demand is dependent on the selling price of the produced item and warranty period. Finally, the managers of this type of manufacturing company will investigate the optimal selling price, optimal production rate, optimal production period, optimal business period and maximum shortage level which maximize the average profit of the system. To find these optimal values, managers may use a hybrid tournamenting QPSO algorithms for solving the corresponding optimization problem (average profit function) of the manufacturing system. On the other hand, on using these algorithms managers of different manufacturing firms utilize computational cost and time with more reliable results because these algorithms give very finer results by taking lesser memory and function evaluations. Hence, managers can use these models in industry.

8 Conclusion

In this study, a production inventory problem has been investigated and formulated the corresponding model with dynamic demand and inventory level-dependent partially backlogged shortages. Due to complexity of the corresponding optimization problem, different variants of metaheuristic optimization techniques AQPSO, GQPSO, AGQPSO, T2-AQPSO, T2-GQPSO and T2-AGQPSO have been applied in order to find the maximum average profit of the system. From the sensitivity analyses, it is observed that the development cost for reducing the production of defective units has a good impact on production process as well as to increase the average profit. Hence, it is concluded that the manufacturer must use development technology to reduce the defective item as well as to increase the average profit. Moreover, after analyzing this model, it is observed that if manufacturer imposes the warranty policy of the product, then its impact directly goes to the customers' demand as well as the average profit. The length of warranty period is dependent on the quality of the products. So, the manufacturer aims to increase the warranty period for increasing demand rate.

From the numerical experiments and results, it can be concluded that the tournament-based hybrid algorithms stabilize the results of a complicated optimization problem than its parent algorithm with lesser computational costs (i.e., memory allocation, function evaluations, CPU times, etc.)

9 Limitations and future research scope

Firstly, the proposed methodologies are based on advanced quantum-behaved particle swarm optimization (AQPSO/AGQPSO/GQPSO) and binary tournamenting process. In these methodologies, initially 4 swarms/populations (teams) are considered. Then in different rounds, populations are updated by (AQPSO/AGQPSO/GQPSO). So, there is no

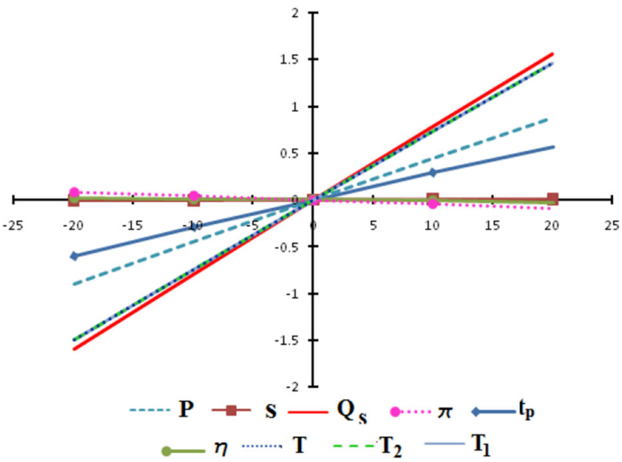

Fig. 9 Sensitivity with respect to A

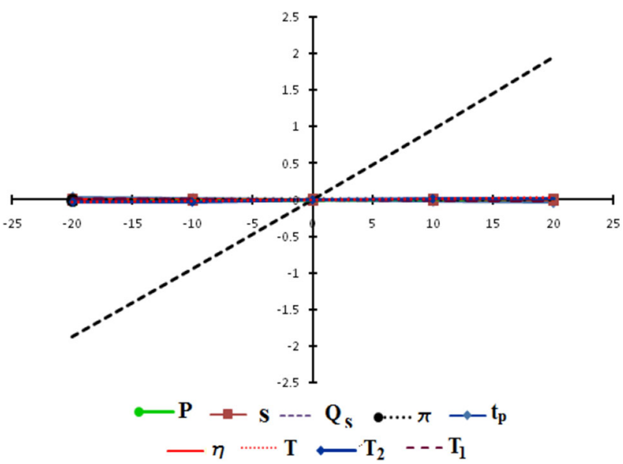

Fig. 10 Sensitivity with respect to c_b

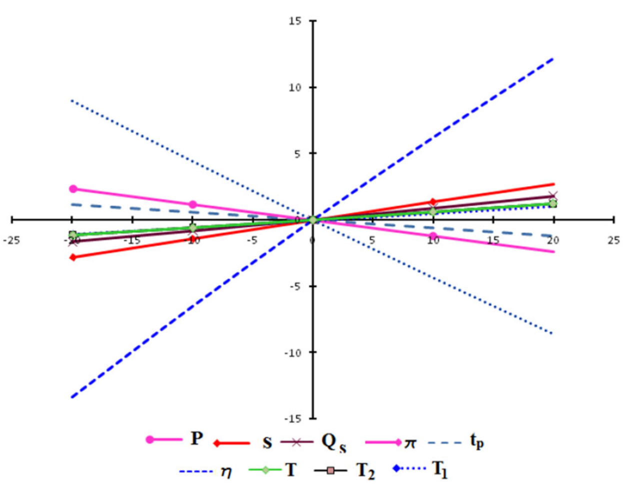

Fig. 11 Sensitivity with respect to C_0

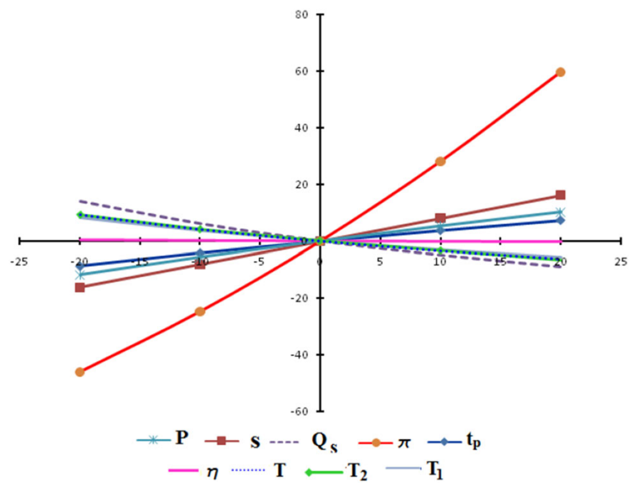

Fig. 12 Sensitivity with respect to β_1

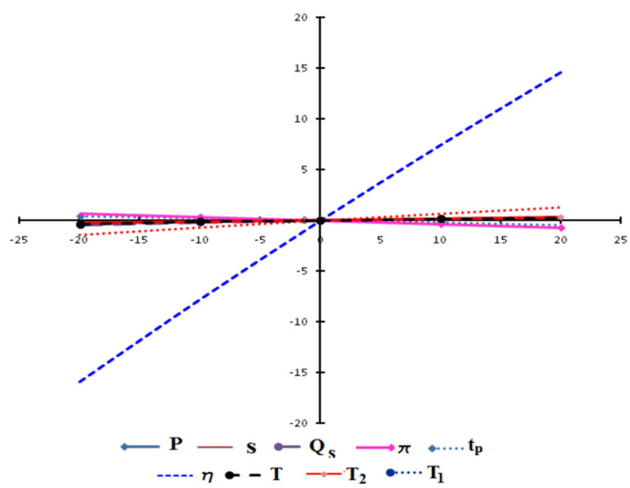

Fig. 13 Sensitivity with respect to θ_0

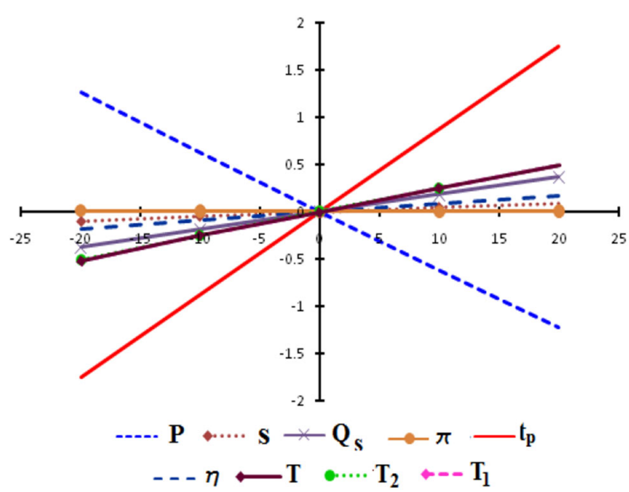

Fig. 14 Sensitivity with respect to w_p

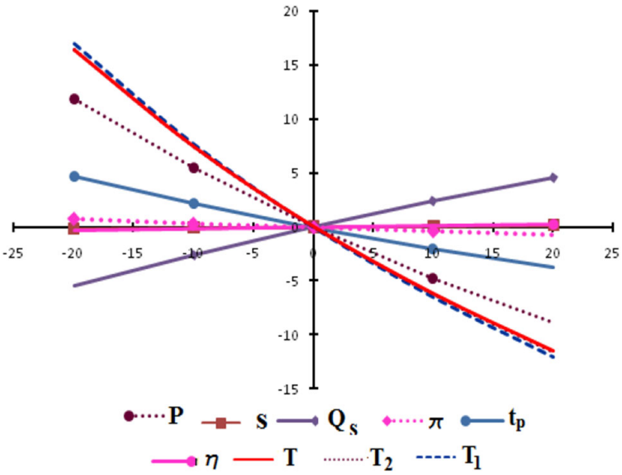

Fig. 15 Sensitivity with respect to h

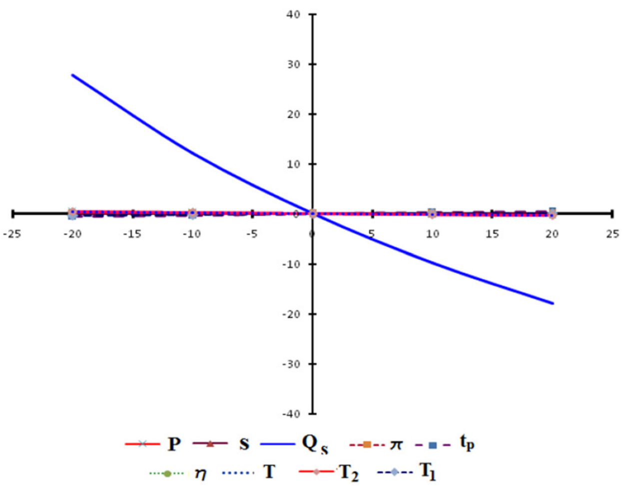

Fig. 16 Sensitivity with respect to α

scope to compare the efficiencies of the methodologies with respect to convergence diagram. Besides, these methodologies work better with suitable swarm size with suitable generations.

Further, this study can be extended by considering several realistic assumptions such as advance payment scheme, time-dependent production rate, time-dependent demand, trade credit facility, inflation, overtime production, etc. This model may also be extended to take interval-valued demand rate, defective rate and different inventory costs. Finally, anyone can extend this model by taking green production process.

Acknowledgements The authors express their sincere thanks to the editor and the anonymous reviewers for their valuable comments and suggestions which have led to a significant improvement of the manuscript. The first author greatly acknowledges the financial support given by Council of Scientific and Industrial Research, New Delhi, India, under Junior Research Fellowship (File no. 09/025(0257)/2018-EMR-I). Also, the second author would like to thank University Grants Commission, India, for providing the Dr. D. S. Kothari Post-Doctoral Fellowship (DSKPDF) through The University of Burdwan for accomplishing this research (Vide Research Grant No.F.4-2/2006 (BSR)/MA/18-19/0023).

Declarations

Conflicts of interest All authors declare that they have no conflict of interest.

Ethical approval There is no involvement of animal.

References

Abad PL (2000) Optimal lot-size for a perishable good under conditions of finite production and partial backordering and lost sale. Comput Ind Eng 38(4):457–465

Abualigah LMQ (2019) Feature selection and enhanced krill herd algorithm for text document clustering. Springer, Berlin, pp 1–165

Abualigah L, Diabat A (2020) A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. Neural Comput Appl, pp 1–24

Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021a) The arithmetic optimization algorithm. Comput Meth Appl Mech Eng 376:113609

Abualigah L, Diabat A (2021b) Advances in sine cosine algorithm: a comprehensive survey. Artificial Int Rev, pp 1–42

Akhtar M, Manna AK, Duary A, Bhunia AK (2020) A hybrid tournament differential evolution algorithm for solving optimization problems and applications. Int J Oper Res. https://doi.org/10.1504/IJOR.2021.10034505

Bhunia AK, Samanta SS (2014) A study of interval metric and its application in multi-objective optimization with interval objectives. Comput Ind Eng 74:169–178

Bhunia AK, Shaikh AA, Cárdenas-Barrón LE (2017) A partially integrated production-inventory model with interval valued inventory costs, variable demand and flexible reliability. Appl Soft Comput 55:491–502

Chakraborty S, Pal M, Nayak PK (2013) Intuitionistic fuzzy optimization technique for Pareto-optimal solution of manufacturing inventory models with shortages. Eur J Oper Res 228:381–387

Chen CK, Lo CC (2006) Optimal production run length for products sold with warranty in an imperfect production system with allowable shortages. Math Comput Modell 44:319–331

Chiu YP (2003) Determining the optimal lot size for the finite production model with random defective rate, the rework process, and backlogging. Eng Optim 35:427–437

Chung CJ (2013) Investigating imperfect process and demand effects on inspection scheduling and supply chain replenishment policy. Comput Ind Eng 64:31–44

Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73

Coelho LS (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37(2):1676–1683

Das BC, Das B, Mondal SK (2017) An integrated production-inventory model with defective item dependent stochastic credit period. Comput Ind Eng 110:255–263

Das SC, Manna AK, Shaikh AA, Zidan AM, Bhunia AK (2020) An application of preservation technology in inventory control system with price dependent demand and partial backlogging. Alex Eng J 59:1359–1369

Das SC, Manna AK, Rahman MS, Shaikh AA, Bhunia AK (2021) An inventory model for non-instantaneous deteriorating items with preservation technology and multiple credit periods-based trade credit financing via particle swarm optimization. Soft Comput. https://doi.org/10.1007/s00500-020-05535-x

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

Duary A, Rahman MS, Shaikh AA, Niaki STA, Bhunia AK (2020) A new hybrid algorithm to solve bound-constrained nonlinear optimization problems. Neural Comput Appl 32(16):12427–12452

Dye C-Y, Yang C-T (2015) Sustainable trade credit and replenishment decisions with credit-linked demand under carbone mission constraints. Eur J Oper Res 244:187–200

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, pp 39–43

García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J Heurist 15(6):617

Giri BC, Jalan AK, Chaudhuri KS (2003) Economic order quantity model with weibull deteriorating distribution, shortage and ramtype demand. Int J Syst Sci 34(4):237–243

Goyal SK, Cárdenas-Barrón LE (2005) Economic production quantity with imperfect production system. Ind Eng J 34(2):33–36

Jaggi CK, Tiwari S, Goel SK (2017) Credit financing in economic ordering policies for non-instantaneous deteriorating items with price dependent demand and two storage facilities. Ann Oper Res 248(1–2):253–280

Jain S, Tiwari S, Cárdenas-Barrón LE, Shaikh AA, Singh SR (2018) A fuzzy imperfect production and repair inventory model with time dependent demand, production and repair rates under inflationary conditions. RAIRO-Oper Res 52(1):217–239

Jamal AMM, Sarker BR, Wang S (1997) An ordering policy for deteriorating items with allowable shortage and permissible delay in payment. J Oper Res Soc 48:826–833

Kumar N, Mahato SK, Bhunia AK (2019) A new QPSO based hybrid algorithm for constrained optimization problems via tournamenting process. Soft Comput. https://doi.org/10.1007/s00500-019-04601-3

Kumar N, Rahman MS, Duary A, Mahato SK, Bhunia AK (2020) A new QPSO based hybrid algorithm for bound-constrained optimisation problem and its application in engineering design problems. Int J Comput Sci Math 12(4):385–412

Kumar N, Shaikh AA, Mahato SK, Bhunia AK (2021a) Applications of new hybrid algorithm based on advanced cuckoo search and adaptive Gaussian quantum behaved particle swarm optimization in solving ordinary differential equations. Exp Syst Appl 172:114646

Kumar N, Shaikh AA, Mahato SK, Bhunia AK (2021b) Development of some techniques for solving system of linear and nonlinear equations via hybrid algorithm. Expert Syst, p e12669

Lee HM, Yao JS (1998) Economic production quantity for fuzzy demand quantity and fuzzy production quantity. Eur J Oper Res 109:203–211

Mallick R, Manna AK, Mondal SK (2018) A supply chain model for imperfect production system with stochastic lead time demand. J Manage Anal 5(4):309–333

Manna AK, Dey JK, Mondal SK (2017a) Imperfect production inventory model with production rate dependent defective rate and advertisement dependent demand. Comput Ind Eng 104:9–22

Manna AK, Das B, Dey JK, Shyamal SK (2017b) Multi-item EPQ model with learning effect on imperfect production over fuzzy-random planning horizon. J Manage Anal 4(1):80–110

Manna AK, Dey JK, Mondal SK (2019) Controlling GHG emission from industrial waste perusal of production inventory model with fuzzy pollution parameters. Int J Syst Sci Oper Logist 6(4):368–393

Manna AK, Dey JK, Mondal SK (2020) Effect of inspection errors on imperfect production inventory model with warranty and price discount dependent demand rate. RAIRO-Oper Res 54:1189–1213

Mishra U, Tijerina-Aguilera J, Tiwari S, Cárdenas-Barrón LE (2021) Production inventory model for controllable deterioration rate with shortages. RAIRO-Oper Res 55:S3–S19

Modak NM, Panda S, Sana SS (2015) Optimal just-in-time buffer inventory for preventive maintenance with imperfect quality items. Tékhne 13(2):135–144

Ouyang L-Y, Chang C-T (2013) Optimal production lot with imperfect production process under permissible delay in payments and complete backlogging. Int J Prod Econ 144:610–617

Rahman MS, Manna AK, Shaikh AA, Bhunia AK (2020) An application of interval differential equation on a production inventory model with interval-valued demand via center-radius optimization technique and Particle Swarm Optimization. Int J Intell Syst 35(8):1280–1326

Rosenblatt MJ, Lee HL (1986) Economic production cycles with imperfect production processes. IIE Trans 18:48–55

Salameh MK, Jaber MY (2000) Economic production quantity model for items with imperfect quality. Int J Prod Econ 64:59–64

Sana SS, Goyal SK, Chauhuri KS (2007) An imperfect production process in a volume flexible inventory model. Int J Prod Econ 105:548–559

Sarker B, Moon I (2011) An EPQ model with inflation in an imperfect production system. App Math Comput 217:6159–6167

Shaikh AA, Mashud AHM, Uddin MS, Khan MAA (2017) Non-instantaneous deterioration inventory model with price and stock dependent demand for fully backlogged shortages under inflation. Int J Forecast 3(2):152–164

Shaikh AA, Khan MAA, Panda GC, Konstantaras I (2019) Price discount facility in an EOQ model for deteriorating items with stock-dependent demand and partial backlogging. Int Trans Oper Res 26(4):1365–1395

Shaikh AA, Cárdenas-Barrón LE, Manna AK, Cespedes-Mota A (2020) An economic production quantity (EPQ) model for a deteriorating item with partial trade credit policy for price dependent demand under inflation and reliability. Yugoslav J Oper Res. https://doi.org/10.2298/YJOR200515036S

Storn R, Price K (1997) Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

Sun J, Feng B, Xu W (2004) Particle swarm optimization with particles having quantum behavior. EEE Congress Comput Intell Evolut Comput 1:325–331

Sun J, Xu W, Feng B (2005) Adaptive parameter control for quantum-behaved particle swarm optimization on individual level. In: 2005 IEEE international conference on systems, man and cybernetics, vol 4, pp 3049–3054

Sun J, Fang W, Palade V, Wu X, Xu W (2011) Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. App Math Comput 218(7):3763–3775

Taleizadeh AA, Hadadpour S, Cárdenas-Barrón LE, Shaikh AA (2017) Warranty and price optimization in a competitive duopoly supply chain with parallel importation. Int J Prod Econ 185:76–88

Taleizadeh AA, Pourreza P, Moon I (2019) Hybrid for an imperfect production system considering product quality and returns under two warranty policies. Appl Soft Comput 75:333–348

Tiwari S, Ahmed W, Sarkar B (2018) Multi-item sustainable green production system under trade-credit and partial backordering. J Clean Prod. https://doi.org/10.1016/j.jclepro.2018.08.181

Wang CH, Sheu SH (2003) Optimal lot size for products under free-repair warranty. Eur J Oper Res 149:131–141

Wilcoxon Signed Rank Test- Statstutor, http://www.statstutor.ac.uk/resources/uploaded/ wilcoxonsignedranktest.pdf

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Wu CC, Chou C-Y, Huang C (2009) Optimal price, warranty length and production rate for free replacement policy in the static demand market. Omega 37(1):29–39

Xi M, Sun J, Xu W (2008) An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. App Math Comput 205(2):751–759

Xu W, Sun J (2005) Adaptive parameter selection of quantum-behaved particle swarm optimization on global level. In: International conference on intelligent computing, Berlin, Heidelberg. pp 420–428

Yeh RH, Ho WT, Teng ST (2005) Optimal production run length for products sold with warranty. Eur J Oper Res 120:575–582